

TP4 : Allocateur Mémoire

Notre Mémoire est de la forme :

taille fit fbh Memoire

Sachant que taille et fit ne sont que les composants de l'allocator_header et fbh est le struct fb qui nous permet d'avoir accès au chaînage des zones libre. Fbh ne correspond donc à aucune zone et le premier fb sera collé au fb.

Pour nous, les zones occupées ne sont pas dans la liste chaînée des zones libre mais ont la même forme. De plus leur pointeur vers le prochain (next) vaut juste NULL dans ce cas.

Nous avons implémenté seulement mem_fit_first() mais avons fait le 5.2)

Nos tests dans memshell :

```
? M
Zone libre, Adresse : 32, Taille : 8144
? a 100
Memoire allouee en 32
? M
Zone occupee, Adresse : 32, Taille : 100
Zone libre, Adresse : 148, Taille : 8028
? a 8029
Echec de l'allocation
? M
Zone occupee, Adresse : 32, Taille : 100
Zone libre, Adresse : 148, Taille : 8028
? a 8028
Finalement la taille sera : 8028
Memoire allouee en 148
? M
Zone occupee, Adresse : 32, Taille : 100
Zone occupee, Adresse : 148, Taille : 8028
? a 1
Echec de l'allocation
? █
```

Donc nous n'avons pas de problème d'allocation dans les cas simples et ne pouvons pas dépasser de la mémoire.

```
? M
Zone libre, Adresse : 32, Taille : 8144
? a 100
Memoire allouee en 32
? M
Zone occupee, Adresse : 32, Taille : 100
Zone libre, Adresse : 148, Taille : 8028
? a 100
Memoire allouee en 148
? M
Zone occupee, Adresse : 32, Taille : 100
Zone occupee, Adresse : 148, Taille : 100
Zone libre, Adresse : 264, Taille : 7912
? l 32
Memoire liberee
? M
Zone libre, Adresse : 32, Taille : 100
Zone occupee, Adresse : 148, Taille : 100
Zone libre, Adresse : 264, Taille : 7912
? a 98
Finalement la taille sera : 100
Memoire allouee en 32
? M
Zone occupee, Adresse : 32, Taille : 100
Zone occupee, Adresse : 148, Taille : 100
Zone libre, Adresse : 264, Taille : 7912
? █
```

Nous avons fait en sorte de ne pas laisser de petits morceaux de memoire lors d'allocations si ces morceaux sont plus petits qu'une balise fb pour éviter les problèmes lors de la lecture d'une zone memoire qui n'aurait pas de fb ou la création d'un fb qui dépasserait sur la zone suivante.

```

? M
Zone libre, Adresse : 32, Taille : 8144
? a100
Memoire allouee en 32
? l32
Memoire liberee
? M
Zone libre, Adresse : 32, Taille : 8144
? l 15
Ce n'est pas une adresse valide
Memoire liberee
? M
Zone libre, Adresse : 32, Taille : 8144
? l95
Ce n'est pas une adresse valide
Memoire liberee
? l32
Cette zone est déjà libre
Memoire liberee
? M
Zone libre, Adresse : 32, Taille : 8144
? 

```

Nous pouvons faire les libérations de mémoires simple sans problème. Par contre si nous demandons une libération à un endroit interdit, il le dit et ne fait rien (contrairement à ce que dit le message “Memoire liberee”).

De plus il nous dit si la zone était déjà libre et ne fait rien.

```

? M
Zone libre, Adresse : 32, Taille : 8144
? a100
Memoire allouee en 32
? a100
Memoire allouee en 148
? M
Zone occupee, Adresse : 32, Taille : 100
Zone occupee, Adresse : 148, Taille : 100
Zone libre, Adresse : 264, Taille : 7912
? l32
Memoire liberee
? M
Zone libre, Adresse : 32, Taille : 100
Zone occupee, Adresse : 148, Taille : 100
Zone libre, Adresse : 264, Taille : 7912
? a100
Finalement la taille sera : 100
Memoire allouee en 32
? M
Zone occupee, Adresse : 32, Taille : 100
Zone occupee, Adresse : 148, Taille : 100
Zone libre, Adresse : 264, Taille : 7912
? l148
Memoire liberee
? M
Zone occupee, Adresse : 32, Taille : 100
Zone libre, Adresse : 148, Taille : 8028

```

```

? a100
Memoire allouee en 148
? a100
Memoire allouee en 264
? M
Zone occupee, Adresse : 32, Taille : 100
Zone occupee, Adresse : 148, Taille : 100
Zone occupee, Adresse : 264, Taille : 100
Zone libre, Adresse : 380, Taille : 7796
? l32
Memoire liberee
? M
Zone libre, Adresse : 32, Taille : 100
Zone occupee, Adresse : 148, Taille : 100
Zone occupee, Adresse : 264, Taille : 100
Zone libre, Adresse : 380, Taille : 7796
? l148
Memoire liberee
? M
Zone libre, Adresse : 32, Taille : 216
Zone occupee, Adresse : 264, Taille : 100
Zone libre, Adresse : 380, Taille : 7796
? l264
Memoire liberee
? M
Zone libre, Adresse : 32, Taille : 8144
? 

```

Dans ce cas nous pouvons voir que nous pouvons libérer des zones et elle se fusionnera automatiquement avec ses zones adjacentes (avant, après ou les deux d’un coup).

5.4) Mem_fit_first sera le meilleur en tant que performance car il n’a pas à vérifier dans toute la mémoire où sera son meilleur emplacement mais juste donne le premier valide.

Mem_fit_best sera le meilleur en terme d’optimisation de la memoire car il laissera la place pour les gros blocs à allouer (pratique car on a pas mis en place d’allocateur qui pourrait séparer les données de l’utilisateur sur plusieurs zones de la memoire pour une meme demande.

Mem_fit_worst nous semble être le pire car il n’a ni les avantages des performances ni les avantages de l’optimisation de mémoire.